

TP9 – Détection d'objets dans une image

L'objectif de ce TP est de détecter et de compter les individus d'une colonie de flamants roses à partir d'une image aérienne (cf. figure 1). Ce comptage, qui permet aux écologues d'étudier l'évolution des populations de flamants roses, se faisait « à la main » jusqu'à une date récente. Une idée simple consiste à positionner N disques dans l'image de telle sorte que le niveau de gris moyen à l'intérieur des disques soit le plus élevé possible. Tous les disques ont le même rayon R . Les coordonnées (réelles) des centres des disques sont tirées selon une loi uniforme. Une fois les N disques positionnés dans l'image, on tente de **maximiser** la somme des niveaux de gris moyens à l'intérieur des disques en procédant comme suit : pour chaque disque traité à tour de rôle, on tire de nouvelles coordonnées ; on accepte la proposition si elle fait croître le niveau de gris moyen. Lancez le script `exercice_0` qui vous est fourni. Quel est manifestement son défaut vis-à-vis de la détection ?

Exercice 1 : détection de flamants roses par champ de Markov

Il semble raisonnable de modéliser un flamant rose par un disque de rayon R fixé, mais pas d'autoriser n'importe quelle intersection entre disques. Faites une copie du script `exercice_0`, de nom `exercice_1`, que vous modifierez de manière à imposer à la distance entre centres de disques d'être supérieure à $\sqrt{2}R$.

Montrez que ce script permet de compter empiriquement le nombre de flamants roses présents dans l'image, en ajoutant des disques à la configuration optimale jusqu'à ce que la totalité des taches claires soient recouvertes par des disques. Si le but est de compter les flamants roses, un défaut manifeste du script `exercice_1` est de devoir fixer le nombre N de disques à l'avance. La notion de *processus ponctuel marqué* permet d'y remédier.

Processus ponctuel marqué

Un disque D de rayon R est caractérisé par les coordonnées $(x, y) \in \mathbb{R}^2$ de son centre C , qui constituent sa *marque*. Pour une configuration $\omega = \{D_1, \dots, D_N\}$ à N individus, l'espace des configurations est \mathbb{R}^{2N} , donc l'espace complet des configurations est $\emptyset \cup \bigcup_{N=1}^{+\infty} \mathbb{R}^{2N}$. Nous définissons un *processus ponctuel marqué* dans cet espace, ce qui nous permet de modéliser le problème par la **minimisation** d'une énergie $U(\omega)$. Cette énergie est la somme d'un terme d'attache aux données $U_{AD}(\omega)$, qui doit être d'autant plus faible que la somme des niveaux de gris moyens est plus élevée, et d'un a priori $U_{AP}(\omega)$ qui pénalise les recouvrements trop importants :

$$U(\omega) = U_{AD}(\omega) + U_{AP}(\omega) = \sum_{1 \leq i \leq N} U_{AD}(D_i) + \beta \sum_{1 \leq i < j \leq N} \delta(d(C_i, C_j) \leq \sqrt{2}R) \quad (1)$$

Dans cette expression, le booléen $\delta(d(C_i, C_j) \leq \sqrt{2}R)$ vaut 1 si et seulement si la distance entre le centre C_i de D_i et le centre C_j de D_j est inférieure à $\sqrt{2}R$. On n'interdit donc pas les recouvrements supérieurs à cette limite, pour lesquels $\delta(d(C_i, C_j) \leq \sqrt{2}R) = 1$: on se contente de les pénaliser, d'autant plus fortement que β est plus élevé. Par ailleurs, le choix du terme d'attache aux données $U_{AD}(\omega)$ est soumis à plusieurs contraintes :

- Comme on cherche à **minimiser** $U(\omega)$, l'énergie individuelle d'attache aux données $U_{AD}(D)$ doit décroître lorsque le niveau de gris moyen $\bar{I}(D)$ à l'intérieur d'un disque D croît.
- L'énergie individuelle $U_{AD}(D)$ d'un disque D bien positionné doit être négative. Sinon, on pourrait faire diminuer $U(\omega)$ en supprimant un tel disque, ce qui serait absurde.
- D'autre part, l'énergie individuelle $U_{AD}(D)$ d'un disque mal positionné doit être positive. Sinon, on pourrait faire diminuer $U(\omega)$ en ajoutant un tel disque, ce qui serait tout aussi absurde.

Une façon de respecter ces trois contraintes consiste à utiliser une fonction *sigmoïde* :

$$U_{AD}(D) = 1 - \frac{2}{1 + \exp \left\{ -\gamma \left[\frac{\bar{I}(D)}{S} - 1 \right] \right\}} \quad (2)$$

où S est un seuil qui indique à partir de quelle valeur limite du niveau de gris moyen $\bar{I}(D)$ on considère qu'un disque est bien positionné. Un exemple de sigmoïde du type (2) est représenté sur le graphe de la figure 1.

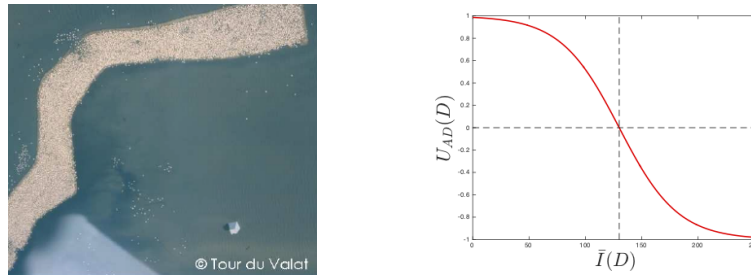


FIGURE 1 – À gauche : image aérienne d’une colonie de flamants roses. À droite : graphe de la sigmoïde de type (2), lorsque $S = 130$ et $\gamma = 5.0$; $U_{AD}(D)$ change de signe en $\bar{I}(D) = S$; la dérivée en ce point vaut $-\gamma/2$.

Optimisation par naissances/morts multiples

Pour **minimiser** l’expression (1) de l’énergie $U(\omega)$, vous allez utiliser un algorithme de *naissances/morts multiples*. Le principe de cet algorithme est d’alterner les phases de naissances, où de nouveaux disques sont ajoutés aléatoirement à la configuration courante, et les phases de morts, où les disques les moins pertinents, au sens de l’énergie, sont supprimés. Cette dynamique, intégrée dans un schéma de recuit simulé, converge vers le minimum global de l’énergie $U(\omega)$. La configuration initiale est l’ensemble vide. L’algorithme consiste à initialiser la température $T \leftarrow T_0$ et le taux de naissances $\lambda \leftarrow \lambda_0$, puis à itérer la séquence suivante :

1. **Naissances.** Tirer le nombre \tilde{N} de nouveaux disques suivant une *loi de Poisson* de moyenne λ :

$$p(\tilde{N}) = e^{-\lambda} \frac{\lambda^{\tilde{N}}}{\tilde{N}!}$$

Ces \tilde{N} nouveaux disques sont positionnés dans l’image de façon aléatoire, suivant une loi uniforme. Ils sont ajoutés à la configuration courante.

2. **Tri des disques.** Une fois la phase de naissances terminée, calculer l’attache aux données $U_{AD}(D)$ de chaque disque D de la configuration courante ω , puis classer les disques par ordre d’énergie décroissante.
3. **Morts.** Pour chaque disque D de la configuration courante ω pris dans cet ordre, calculer la probabilité :

$$p(D|\omega) = \frac{\lambda}{\lambda + \exp \left\{ \frac{U(\omega \setminus \{D\}) - U(\omega)}{T} \right\}}$$

et supprimer D , c’est-à-dire faire $\omega \leftarrow \omega \setminus \{D\}$, avec la probabilité $p(D|\omega)$.

4. **Test de convergence.** Si le processus n’a pas convergé, c’est-à-dire si la configuration après l’étape de morts a changé par rapport à l’itération précédente, modifier la température T et le taux de naissances λ en les multipliant par un même coefficient de décroissance $0 < \alpha < 1$. Retourner à l’étape de naissances.

Exercice 2 : détection de flamants roses par processus ponctuel marqué

Faites une copie du script `exercice_1`, de nom `exercice_2`, que vous modifierez de manière à détecter les flamants roses selon l’algorithme décrit ci-dessus, en choisissant par exemple les valeurs suivantes pour les paramètres du modèle : $\beta = 1.0$, $S = 130$, $\gamma = 5.0$, $T_0 = 0.1$, $\lambda_0 = 100.0$ et $\alpha = 0.99$. La condition d’arrêt de l’algorithme (étape 4) pourra être simplifiée, en fixant par avance le nombre d’itérations à `q_max = 120`. Par ailleurs, le graphique situé à droite (`subplot(1,2,2)`) devra montrer l’évolution de l’énergie $U(\omega)$ de la configuration courante en fonction de q (l’échelle de cette courbe pourra être fixée à `[0 q_max -400 0]`).

Indication La fonction `poissrnd` de Matlab permet de tirer un nombre aléatoire suivant une loi de Poisson.

Jouez sur les paramètres, de manière à compter « au plus juste » le nombre N_{lim} d’individus de la colonie de l’image `colonie.png`. Relancez le script `exercice_1`, en fixant N à la valeur N_{lim} . Comparez les résultats.